Temperature Alert System

1. Theoretical Part

For this project I chose to implement a quite simple circuit: a temperature alert system. Inherently, it makes you aware that your house is at an unhealthy temperature or even the greenhouse of some very neat plants. [1]

About Arduino uno

Arduino is a small microcontroller board with a USB plug to connect to your computer and a number of connection sockets that can be wired up to external electronics, such as motors, relays, light sensors, laser diodes, loudspeakers, microphones, etc. Arduinos can be powered either through the USB connection from the computer or from a 9V battery. They can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently.

Although Arduino is an open-source design for a microcontroller interface board, it is actually rather more than that, as it encompasses the software development tools that you need to program an Arduino board, as well as the board itself. There is a large community of construction, programming, electronics, and even art enthusiasts willing to share their expertise and experience on the Internet.

The heart of our Arduino is a microcontroller. Practically everything else on the board is concerned with providing the board with power and allowing it to communicate with your computer.

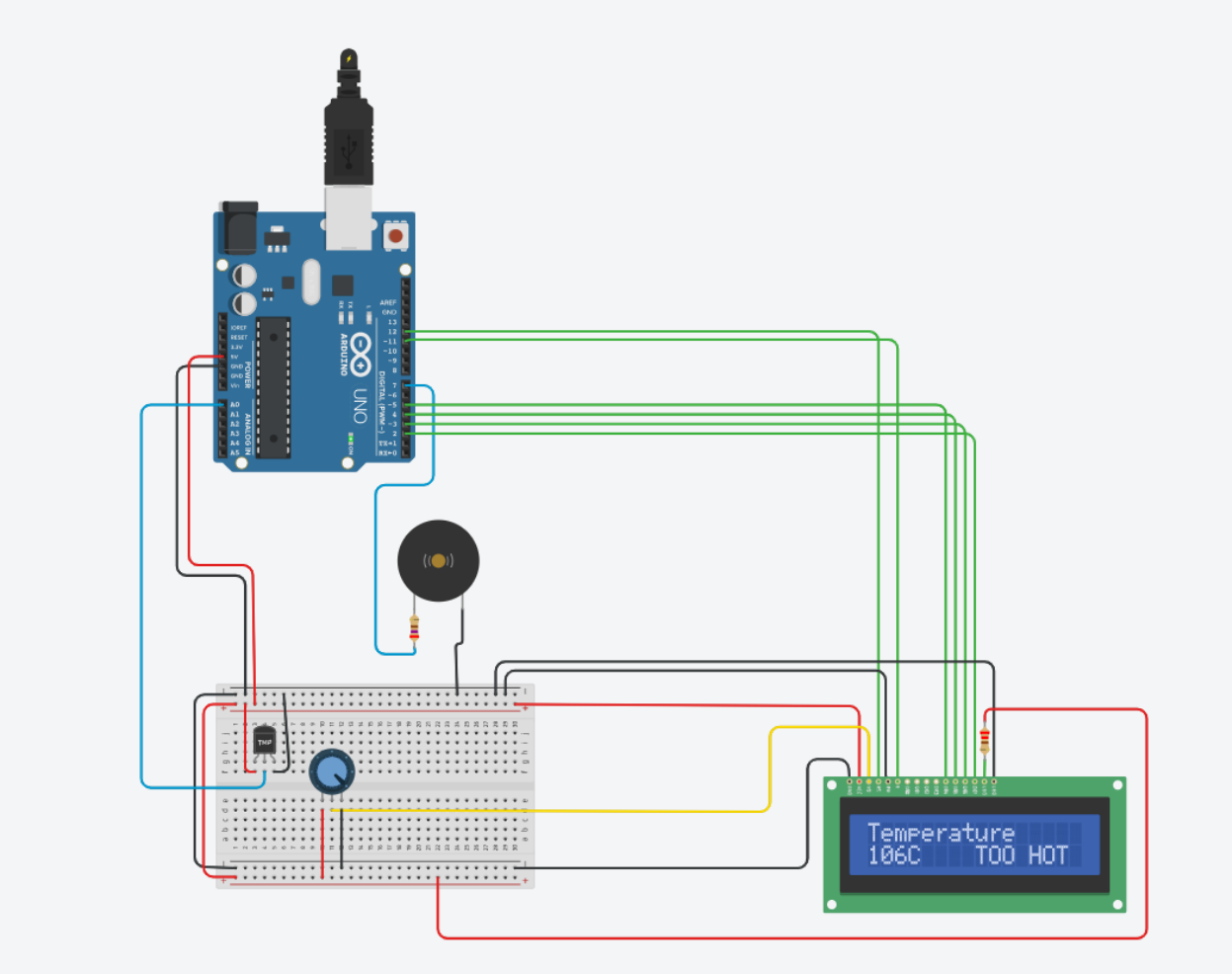

Figure 1. Tinkercad circuit of the temperature alert system[1]

Table 1. Components used

NR.	NAME OF THE COMPONENT	VALUE (IF NEEDED)
1	ARDUINO UNO	-
2	BREADBOARD	-
3	LCD 16x3	-
4	RESISTOR	270Ohm
5	RESISTOR	220Ohm
6	PIEZO BUZZER	-
7	TEMPERATURE SENSOR [TMP36]	-
8	POTENTIOMETER	250KOhm

2. Electrical circuit (Tinkercad)

The circuit is built using an Arduino Uno board, a small and simple circuit on the breadboard with a TEM36 temperature sensor, a potentiometer and a resistor, an 16x2 LCD screen with 16 pins and a buzzer.

The LCD screen is connected to the Arduino Uno and a potentiometer. The said potentiometer is used to dim, brighten, or turn off the LCD screen, depending on your needs. This circuit element can be removed if necessary by connecting the VD pin of the LCD to ground. Eight of the pins are data lines (pins 7–14), two are for power and ground (pins 1 and 16), three are used to control the operation of LCD (pins 4–6), and one is used to adjust the LCD screen brightness (pin 3). The remaining two pins (15 and 16) power the backlight. The pins I used to connect my LCD can be seen in

The temperature sensor I used is the TMP36, the only one available in Tinkercad. This sensor provides a highly precise temperature in centigrade. Most importantly, it produces output in dc voltage that we can measure easily with the help of any bare metal microcontrollers such as Arduino Uno, STM32F4, PIC16F877A. On top of that, Celsius's temperature and an output voltage change linearly which makes it easy to compensate temperature/Voltage variations. Having a linear relationship is helpful. Because we will not require any external calibration circuit. Furthermore, it offers a very low output impedance. In short, it is very easy to interface this sensor with ADCs or microcontrollers having built-in ADCs. These devices can handle temperature ranges of -40°C to 150°C. [4] With this in mind and the following calculations, I managed to convert volts into degrees:

```
celsius = map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
```

As the temperature range is very wide, I chose to alert the user when the temperature is too cold (below 10 degrees Celsius) and when it is too hot (over 26 degrees Celsius). When one of the two cases happen, the buzzer will make a series of strident musical notes.

In order to implement this circuit I used tinkercad.com. Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding. It's used by teachers, kids, hobbyists, and designers to imagine, design, and make anything. [3]

Link to the tinkercad project: https://www.tinkercad.com/things/0pw5ie26yvJ

3. Arduino IDE

```
// include the library code:
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //connecting the pins rs,en,d4,d5,d6,d7 to the arduino at pin 12 11 5 4 3 2
int celsius; //declare a function celsius as an integer

void setup()
{
	pinMode(A0,INPUT);
	pinMode(7,OUTPUT);
	lcd.begin(16, 2); //lcd size is 16x2 // Print a message to the LCD.

}

void loop()
{
	celsius = map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125); //map to obtain temperature mathematically.Meaning 0 = -40degrees and 1023 = 125degrees

	lcd.setCursor(0,0); //cursor set to the first pixel of the lcd.
	lcd.print("Temperature");
	Serial.println("Temperature"); //print the message at the serial monitor
	lcd.setCursor(0,1);//cursor set to the second line first pixel
	lcd.print(celsius); //prints the celsius output from the analog read onto the lcd at 0,1
	lcd.print("C"); //print alphabet “c”
	Serial.println(celsius); //print the message at the serial monitor
	delay(500); //reading refreshes every 0.5 second
	lcd.clear(); //clears the lcd

	if(celsius>30)
	{
		digitalWrite(7,1); //starts the buzzer when the temperature is over 30
		tone(7, 261, 3830);  // play note c for 200 ms
		delay(200);

		noTone(7); // turn off tone function
		tone(7, 392, 2550); // play note g for 500 ms
		delay(500);

		noTone(7); // turn off tone function
		tone(7, 261, 3830); // play note f on pin 6 for 400 ms
		delay(400);
```

```
        noTone(7); // turn off tone function
        tone(7, 349, 2864); // play note c on pin 6 for 600 ms
                delay(600);

                lcd.setCursor(8,1); //cursor set to the 9th pixel of the lcd.
                lcd.print("TOO HOT"); //print the message at the serial monitor
        }
                else
                        {
                  if(celsius<30 && celsius>=15)
                  {
                        digitalWrite(7,0); //keeps buzzer off
                        lcd.setCursor(8,1); //cursor set to the second line 9th pixel of the
lcd.
                        lcd.print("Normal");//print the message at the serial monitor
                  }
                  else
                  {
                     if(celsius<15 && celsius>=10)
                     {
                          digitalWrite(7,0); //keeps buzzer off
                                lcd.setCursor(8,1); //cursor set to the second line 9th
pixel of the lcd.
                          lcd.print("Cold"); //print the message at the serial monitor
                     }
                       else
                       {
                         if(celsius<10)
                        {
                             digitalWrite(7,1); //starts the buzzer when the temperature is
under 10
                             tone(7, 294, 3400);  // play note d for 200 ms
                            delay(200);

                            noTone(7); // turn off tone function
                            tone(7, 329, 3038); // play note e for 500 ms
                            delay(500);

                            noTone(7); // turn off tone function
                            tone(7, 493, 2028); // play note b on pin 6 for 400 ms
                            delay(400);

                            noTone(7); // turn off tone function
                            tone(7, 523, 1912); // play note C on pin 6 for 600 ms
                            delay(600);
```

```
                    lcd.setCursor(8,1); //cursor set to the 9th pixel of the lcd.
                    lcd.print("TOO COLD"); //print the message at the serial
monitor
                }
                }
            }
        }
}
```

Reference list

[1] (Formagio, 2020)
[2] (Szolga, 2020)
[3] (About, 2021)
[4] (TMP36 Low Voltage Temperature Sensor, 2021)